

# Ready...Set...Start Your Containers

61% of container technology adopters expect more than 50% of their existing and new applications to be packaged on containers over the next two years.

## Author

Siva Sreeraman VP,  
CTO and Modernization Tribe Leader

In the decades past, developers faced many errors when porting applications created for a specific computing environment. Configuration differences such as versions of compilers, loaders, runtime libraries, middleware and operating system in new environments created incompatibility and unreliability, and led to undesired increases in project effort, cost and timelines.

Containers provide an elegant solution to this problem. Each container leverages a shared operating system kernel and encapsulates everything needed to run an application (application code, dependencies, environment variables, application runtimes, libraries, system tools, etc.) in an isolated and executable unit. Differences in operating system distributions and underlying infrastructure configurations are thus abstracted away, allowing application programs to run correctly and identically even when deployed to different environments.

## How we got here

Containerization originated in 2001 as a project allowing several general-purpose Linux servers to run on a single box with autonomy and security. Subsequent projects at IBM, Red Hat and Docker moved this technology forward over the years. In 2014, Google launched its container orchestration platform Kubernetes (K8s) and declared that it started over 2 billion containers on a weekly basis. In 2020, the Cloud Native Container Foundation released data that indicated an overwhelming preference for Kubernetes among companies that used containers in production.

Many organizations today decouple their complex monolithic applications into modular, manageable microservices packaged in containers that can be linked together. Container orchestrators such as Kubernetes further automate installation, deployment, scaling and management of containerized application workloads on clusters, perform logging, debugging, version updates and more.

## How it works

Containers in Kubernetes, the most widespread container orchestrator, are implemented using Linux kernel features called namespaces and cgroups (control groups). Namespaces limit what system resources (CPU, memory, disk I/O, network traffic, etc.) a containerized process or a set of processes can see. Cgroups limit the system resources that a containerized process or a set of processes can use. Together, they enable strong isolation, preventing containers from gaining control over each other's resources.

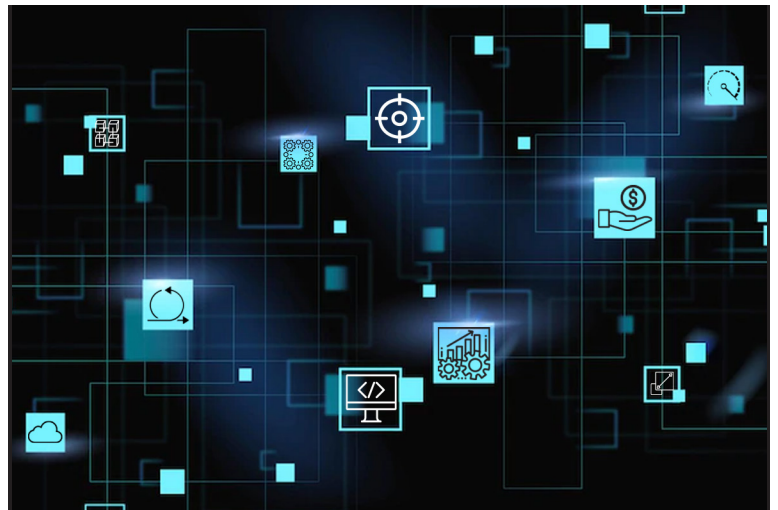
Containers are grouped into deployable computing units called pods, which contain shared network and storage resources and specifications on how to run the containers. Pods run on nodes – physical or virtual machines containing a set of CPU and RAM resources. Nodes are managed by the container orchestration layer and pool together into more powerful machines called clusters. Clusters distribute work among individual nodes as needed to execute programs. If any nodes are attached or removed, the cluster manages this, and it remains transparent to the program.

## Advantages

Containers appeal to the software development community because of the agility, uniformity and portability they provide in creating and deploying applications and their consistent performance of code execution irrespective of the run time environment – a ‘write once, run anywhere’ approach across different infrastructures, on-premise or in the cloud. Container images can be quickly rolled back in case of any issues observed. They can be rapidly spun up, adding business functionality and scalability on demand, and torn down, reducing resource usage and infrastructure costs.

Since containers do not need to run a full operating system and share the host machine’s operating system kernel with each other, they are lightweight and do not have the same resource utilization needs as virtual machines do. Containers are faster to start up, drive higher server efficiencies and reduce server and licensing costs.

Containers allow developers to focus on business functionality and not worry about the underlying configurations of applications. A consistent and short deployment process enables faster delivery of new applications. 75% of companies using containers achieved a moderate to significant increase in application delivery speed.



A great benefit of isolating applications into containers is the inherent security provided. As images are the building blocks of containers, maliciously introduced code as well as unnecessary components can be prevented from entering containers by using trusted image registries, enhanced access control methods and strict policies applied to both accounts and operations. Whenever changes are made to container configurations, or containers started, auditability must be implemented.

## Challenges

Though containers solve a lot of security problems compared to traditional virtualization methods, they also introduce new security challenges. As the Kubernetes cluster attack surface vector area is so large and increasing exponentially – there are layers upon layers of images that span thousands of machines and services – this has provided many opportunities for cybercriminals to launch coordinated attacks on Kubernetes to access company networks by taking advantage of any misconfigurations.

Recent attacks have introduced cryptojacking, wherein an organization’s vast compute resources on the cloud are unsuspectingly diverted towards mining cryptocurrency. As Kubernetes manages other machines and networks, enterprises should continuously strengthen their security postures and take proactive measures to defend themselves.

Though container cluster managers such as Docker Swarm and Apache Mesos have enabled developers to build, ship and schedule multi-container applications, and access, share and consume container pools through APIs, container scaling is still evolving. Container orchestration tools and container cluster managers have not fully integrated with each other. Cluster managers today are not able to provide security at enterprise-class levels and a common set of standards is lacking.

# Containerization best practices

## Current best practices for container operations include:

- Avoid privileged containers, which could allow attackers to bypass container security features and gain access to all the devices of the host machine
- Statelessness, i.e., storing any state/persistent data externally and thereby permitting graceful shutdown of containers and no data losses
- Keep containers immutable, i.e., no modifications over their life to apply any application updates, security patches or configuration changes, thus allowing for safe and identical deployments in every environment
- Securely manage passwords, secrets and roles on a per-pod basis and frequently rolling credentials
- Do not use any backdoors or gateways that can provide ingress mechanisms to attackers
- Implement quotas so that resources are not exhausted and any outage is restricted to the defined constraints
- Update clusters to use recent major versions of Kubernetes and applying security patches consistently
- Use cloud-managed Kubernetes services where possible to lower the degree of difficulty in self-managing on-premises Kubernetes installations

## In conclusion

Despite challenges, containers present many benefits and offer enterprises an attractive choice for software application development. 61% of container technology adopters expect more than 50% of their existing and new applications to be packaged on containers over the next two years. By 2026, Gartner estimates that 90% of global organizations will be running containerized applications in production.

The usage of managed public cloud Container-as-a-Service (CaaS) such as Amazon Web Services (AWS), Elastic Kubernetes Service (EKS), Microsoft Azure Kubernetes Service (AKS) and Google Kubernetes Engine (GKE) is widespread among enterprises today. Container-based Platform-as-a-Service (PaaS) offerings such as Google Cloud Anthos, Red Hat OpenShift, VMware Tanzu Application Service and SUSE Rancher are also prevalent. Lightweight Kubernetes distributions (with half the memory needed for K8s and smaller binary sizes) like SUSE Rancher K3s and Mirantis K0s can be seen in Edge, Internet of Things and Reduced Instruction Set Computing applications.

While the introduction of containers may add some vulnerabilities, the speed, efficiency and savings they provide in return are well worth the easily managed risk. Thanks to these considerable benefits, container technology will continue to be a foundational element of the enterprise software technology stack over the coming years. Companies should continue to invest in and utilize containerization in their digital transformation journeys.

## About Mphasis

Mphasis' purpose is to be the "Driver in Driverless Car" for Global Enterprises by applying next-generation design, architecture and engineering services, to deliver scalable and sustainable software and technology solutions. Customer centricity is foundational to Mphasis, and is reflected in the Mphasis' Front2Back™ Transformation approach. Front2Back™ uses the exponential power of cloud and cognitive to provide hyper-personalized ( $C = X2C_m = 1$ ) digital experience to clients and their end customers. Mphasis' Service Transformation approach helps 'shrink the core' through the application of digital technologies across legacy environments within an enterprise, enabling businesses to stay ahead in a changing world. Mphasis' core reference architectures and tools, speed and innovation with domain expertise and specialization, combined with an integrated sustainability and purpose-led approach across its operations and solutions are key to building strong relationships with marquee clients.

[Click here](#) to know more. (BSE: 526299; NSE: MPHASIS)

For more information, contact: [marketinginfo.m@mphasis.com](mailto:marketinginfo.m@mphasis.com)

**USA**  
460 Park Avenue South  
Suite #1101  
New York, NY 10016, USA  
Tel.: +1 646 424 5145

**UK**  
Mphasis UK Limited  
1 Ropemaker Street, London  
EC2Y 9HT, United Kingdom  
T : +44 020 7153 1327

**INDIA**  
Bagmane World Technology Center  
Marathahalli Ring Road  
Doddanakundhi Village  
Mahadevapura  
Bangalore 560 048, India  
Tel.: +91 80 3352 5000

